IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: BRIAN C. BARNES, GEOFFREY S. STRONGIN, RODNEY W. SCHMIDT	Group Art Unit: 2126
Serial No.: 10/044,667	Examiner: VAN H. NGUYEN
Filed: JANUARY 11, 2002	Conf. No.: 9420
For: METHOD AND APPARATUS FOR LINEAR ADDRESS BASED PAGE LEVEL SECURITY SCHEME TO DETERMINE CURRENT SECURITY CONTEXT	Atty. Dkt.: 2000.057000/TT4090
	CUSTOMER NO.: 23720

SECOND AMENDED APPEAL BRIEF

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

On January 30, 2006, Appellants filed a Notice of Appeal in response to a Final Office Action dated September 30, 2005, issued in connection with the above-identified application. In support of the appeal, Appellants had submitted an Appeal Brief to the Board of Patent Appeals and Interferences. Appellants had also filed an Amended Appeal Brief on August 24, 2006. Subsequently, the Office mailed a Notification of Non-Compliant Appeal Brief on October 2, 2006. In response, Appellants hereby file this Second Amended Appeal Brief.

Since the Notification of Non-Compliant Appeal Brief was mailed on October 2, 2006, one-month date for responding is November 2, 2006. Since this Amended Appeal Brief is being filed on November 2, 2006, this paper if filed timely.

If an extension of time is required to enable this paper to be timely filed and there is no separate Petition for Extension of Time filed herewith, this paper is to be construed as also constituting a Petition for Extension of Time Under 37 CFR § 1.136(a) for a period of time sufficient to enable this document to be timely filed.

No other fee is believed to be due in connection with the filing of this document. However, should any fee under 37 C.F.R. §§ 1.16 to 1.21 be deemed necessary for any reason relating to this document, the Commissioner is hereby authorized to deduct said fee from Williams, Morgan & Amerson, P.C. Deposit Account No. 50-0786/2000.057000.

I. REAL PARTY IN INTEREST

The present application is owned by Advanced Micro Devices, Inc.

II. RELATED APPEALS AND INTERFERENCES

Appellants are not aware of any related appeals and/or interferences that might affect the outcome of this proceeding.

III. STATUS OF CLAIMS

Claims 1-26 remain pending in this application. The Examiner rejected claims 1-23, 25 and 26. Additionally, the Examiner objected to claim 24, which has been canceled and removed from the Claims Appendix of the present Second Amended Appeal Brief.

IV. STATUS OF AMENDMENTS

After the Final Rejection, amendments to claims 4, 8, 15, and 20, were made and, Appellants respectfully acknowledge that, for purposes of this Appeal, have been entered by the Examiner.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Embodiments of the present invention provide for memory access using security access systems. Embodiments of the present invention provide for a multiple memory access table system to provide security during a memory access initiated by one or more processors in a computer system. Embodiments of the present invention also provide a virtual memory access system (200) that utilizes a primary virtual memory access table (410) and a secondary memory access table (430), which results in increased security during memory accesses. Embodiments of the present invention provide for performing memory access and/or obtaining security attributes based upon virtual addressing. *See* Specification, page 8, line 11-18; Figures 2, 3 and 7.

In one aspect of the present invention, a method is provided for performing a virtual address based memory access (640) using targeted security. A software object (350) is executed. A security level (100) for the software object is established. A virtual address based memory access (640) is performed using the security level (100). Performing the virtual address based memory access (640) includes using a secondary table (430) and at least one virtual memory table (410). *See* Specification, page 17, line 19-page 18, line 9; Figures 1, 3, 6, and 7

In another aspect of the present invention, a method is provided for performing a virtual address based memory access (640) using targeted security. A software object (350) is executed. A security level (100) for the software object (350) is established. A secondary table (430) is established. A memory access request is received based upon the executing of the software object (350). At least one security level (100) that corresponds to a segment in the secondary table (430) is determined based upon a virtual address. A portion of a memory (340) based upon the security level (100) and the virtual address is accessed. Accessing the portion of the

memory (340) includes using the secondary table (430) and at least one virtual memory table (410). *See* Specification, page 18, line 11-page 19, line 23; Figures 1, 3, 6, and 7.

In another aspect of the present invention, an apparatus is provided for performing a virtual address based memory access (640) using targeted security. The apparatus includes means (310) for executing a software object (350). The apparatus also includes means (310) for establishing a security level (100) for the software object. The apparatus includes means (320) for performing a virtual address based memory access using the security level (100). The means (320) for performing the virtual address based memory access includes means for using a secondary table (430) and at least one virtual memory table (410). *See* Specification, page 9, line 8-page 10, line 23; Figures 1, 3, and 6.

In another aspect of the present invention, an apparatus (210) is provided for performing a virtual address based memory access (640) using targeted security. The apparatus (210) includes a processor (310) that is coupled to a bus (315, 325). The apparatus (210) also includes means (315) for coupling at least one software object to the processor (310). The apparatus (210) also includes a memory unit (340) and a memory access interface (320) that is coupled to the bus (315) and the memory unit (340). The memory access interface (320) is adapted to provide the processor (310) a virtual address based access of at least a portion of the memory unit (340) based upon at least one security level (100) in response to the processor (310) executing the software object. The processor (310) is adapted to use a secondary table (430) and at least one virtual memory table (410) to perform the virtual address based access. *See* Specification, page 9, line 8-page 10, line 23; Figures 1, 2, 3, and 6.

In yet another aspect of the present invention, a computer readable program storage device (200) is provided for performing a virtual address based memory access (640) using targeted security The computer readable program storage device (200) is encoded with instructions that, when executed by a computer (210), performs a method, that includes: executing a software object (350); establishing a security level (100) for the software object (350), and performing a virtual address based memory access (640) using the security level (100). Performing the virtual address based memory access (640) includes using a secondary table (430) and at least one virtual memory table (410). *See* Specification, page 9, lines 8-13; page 8, lines 11-18; Figure 1, 2, 3, 6, and 7.

In another aspect of the present invention, a method is provided for performing a virtual address based memory access (640) using targeted security. A software object (350) is executed. A security level (100) for the software object (350) is established. A secondary table (430) is established. Establishing the secondary table (430) includes: dividing a physical memory (345) into a plurality of segments; determining at least one of the segment to omit from the secondary table (430) and at least one un-omitted segment, assigning a default security level (100) to the omitted segment; assigning a security level (100) to the un-omitted segment; and correlating at least one assigned segment with a virtual memory location. A virtual address based memory access (640) is performed using at least one of the security levels (100). Performing the virtual based address memory access (640) includes using the secondary table (430) and at least one virtual memory table (410). The function of the object (350) is executed based upon the virtual address based memory access. *See* Specification, page 20, line 1-page 21, line 11; Figures 1, 3, 6, and 8.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Whether claims 4-5, 8-11, 15 and 20-21 are unpatentable under 35 U.S.C. 112, as being indefinite;

2. Whether claims 1-4, 6-20, 22, 23, 5 and 26 are unpatentable under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,745,308 ***(McKee)***.

VII. ARGUMENT

The Examiner issued a Notice of Non-compliant Appeal Brief mailed October 2, 2006. The Examiner asserted that the Claims Appendix included an objected claim (claim 24). In the interest of expediency, Appellants have canceled claim 24 and removed the claim from the Claims Appendix. Appellants reserve the right to pursue claim 24 in a continuing application. In light of removing claim 24 from the Claims Appendix, Appellants respectfully assert that the Claim Appendix is in compliance with 37 C.F.R. 41.37(c)(1)(viii)). Accordingly, the present Second Amended Appeal Brief is in compliance with all requirements of 37 C.F.R. 41.37(c).

The Examiner's objections in the Notice of Non-compliant Appeal Brief mailed July 24, 2006 were addressed in the present Amended Appeal Brief filed on August 24, 2006. Various modifications to the Summary section had been made. Appellants respectfully assert that the Amended Appeal Brief complies with all requirements of 37 C.F.R. 41.37(c).

The present invention is directed to executing a software object and establishing a security level for the software object. The present invention is also directed to performing a virtual address based memory access using the security level, wherein the access includes using a secondary table and at least one virtual memory table. The Examiner relies heavily upon U.S.

Patent No. 6.745,307 (***McKee***). The Examiner equates a translation look-aside buffer (TLB) cited in ***McKee*** with the secondary table of the claims, in order to argue anticipation of the claims of the present invention. The Examiner has misapplied the disclosure of ***McKee***. One fundamental flaw in the Examiner's argument is that ***McKee*** discloses that data is entered into the TLB as a result of having performed the translation of the virtual memory addressing to physical memory address, which is in direct opposition to the Examiner's argument that the TLB is used to perform the translation. This portion of the disclosure of ***McKee*** used by the Examiner is actually opposite to claims of the present invention that performs the virtual address based memory access using a secondary table and at least one virtual memory table. In other words, the virtual address based memory access is not performed using the TLB and a virtual page table (602) of ***McKee*** but instead, data is written into the TLB as a result of having performed the translation of the virtual memory access into physical memory address. Therefore, the Examiner is entirely incorrect in asserting that the ***McKee*** teaches or anticipates all of the elements of claims of the present invention.

The specific claims of the present invention are discussed below.

A. Claims 1-11 Are Not Invalid As A Result Of Mental Process And Are In Compliance With 35 U.S.C 101 And Claims 4-5, 8-11, 15 and 20-21 Are Not Unpatentanble Under 35 U.S.C. 112, as being indefinite

The Examiner objected to claims 1-11 based upon mental process. Appellants respectfully disagree. Appellants respectfully assert that claims 1-11 recite a novel process that is statutory subject matter. Appellants respectfully asserts that contrary to Examiner's assertion (in paragraph 2 of the Final Office Action dated September 30, 2005), 35 U.S.C. 101 does not require that a process or method must recite "a computer implemented method" to be statutory

subject matter. The requirements under 35 U.S.C. 101 call for a novel, non-obvious process, which is the subject matter of claims 1-11. Claims 1-11 call for a novel, non-obvious method for performing a memory access. Claims 1-11 need not recite "a computer implemented method" in order to be patentable statutory subject matter. Therefore, claims 1-11 contain allowable subject matter, and thereby are allowable. Accordingly, the Examiner erred in objecting to claims 1-11 and this objection should be reversed.

Further, amendments to claims 4, 8, 15, and 20 (in the Response to Final Office Action) have been entered and therefore, the rejection of claims 4-5, 8-11, 15 and 20-21 under 35 U.S.C. 112, second paragraph, as being indefinite are now moot. The amendments to claims 4, 8, 15, and 20 are not indefinite, therefore claims 5, 9-11, and 21, which are dependent from the rejected claims are also allowable for at least the same reasons. Accordingly, claims 4-5, 8-11, 15 and 20-21 are allowable.

B. Claims 1-4, 6-20, 22, 23, 25 and 26 Are Not anticipated Under 35 U.S.C. § 102(e) by U.S. Patent No. 6,745,307 (*McKee*)

The Examiner rejected claims 1-4, 6-20, 22, 23, 25 and 26 under 35 U.S. 102(e) as being anticipated by U.S. Patent No. 6,745,307 ***(McKee)***. The Examiner has erred in placing forward this rejection; therefore, Appellants respectfully traverse this rejection.

In the Final Office Action dated September 30, 2005, the Examiner argued that ***McKee*** teaches performing a virtual address based memory access. The Examiner cites, among other passages, the translation of the virtual memory address 502 to a physical memory address 508 disclosed on column 7, lines 35 onwards. However, this disclosure is related to performing the translation using a virtual memory table 514. The use of the virtual memory table as described

below, does not anticipate the use of a secondary table, as well as a virtual memory table called for by claims of the present invention. The portion cited by the Examiner relates to performing the physical virtual memory address to physical memory address translation entirely by a processor without operating system intervention. Further, the usage of the TLB and the virtual page address of ***McKee*** does not teach, disclose, or suggest performing a virtual address memory access using a secondary table and at least one virtual memory table as called for by claim 1 of the present invention.

McKee simply does not teach, disclose, or suggest all of the elements of claim 1 of the present invention. For example, the virtual memory access called for by claim 1 of the present invention calls for performing a virtual address based memory access that is based on a secondary table and at least one virtual memory table. The Examiner asserted that this element is anticipated by the usage of the TLB and the virtual page address 505 of ***McKee.*** The Examiner asserted that this disclosure anticipated the use of the secondary and of virtual memory table of claim 4, as well as claim 1. Appellants respectfully disagree. Appellants respectfully assert that the use of the secondary table as well as the virtual memory table is not anticipated by the TLB and the virtual page address 505 of ***McKee***.

The TLB is a translation look-aside buffer. The TLB contains data that is actually written by an operating system. For example, ***McKee*** discloses that the virtual page table entry 602 contains additional fields from which information required for a TLB entry can be retrieved. See col. 8, line 66 col. 9, line 1. ***McKee*** discloses that if the operating system successfully translates the virtual memory address into a physical memory address, that translation, both as a virtual page table entry and as a TLB entry, is inserted into the TLB. See col. 9, lines 1-4. This disclosure makes it clear that data is entered into the TLB as a result of translating virtual

memory address into physical memory address, and not used to perform a virtual address based memory access. In other words, the virtual address based memory access is not performed using the TLB and the virtual page table 602, contrary to the Examiner's assertions. In fact, the above cited passage in ***McKee*** makes it abundantly clear that the prior art discloses that memory access is performed prior to writing data into the TLB *i.e.*, the virtual memory address being translated into a physical memory address. Subsequently, that information is then entered into the TLB. Therefore, it is erroneous to argue that the virtual address memory access in ***McKee*** is performed using two entities, such as the TLB and the virtual address table. Hence, the disclosure of the memory access in ***McKee*** is in stark contrast with the virtual address based memory access called for by claim 1 of the present invention, which calls for using a secondary table and a virtual memory table. Therefore, Appellants respectfully assert that the usage by ***McKee*** of the TLB and the virtual page table do not equate, anticipate or make obvious the element of the virtual address memory based access called for by claim 1, which calls for using a secondary table and a virtual table access.

In fact, ***McKee*** simply does not disclose a secondary table. The Examiner's usage of the TLB is erroneous since data is written into the TLB, wherein the virtual memory access called for by claim 1 of present invention uses information in the secondary table as well as a virtual memory table. In other words, ***McKee*** does not anticipate the subject matter of virtual address memory access using a secondary table and a virtual table access. Therefore, claim 1 of the present invention is not taught, disclosed or suggested by ***McKee***. Accordingly, claim 1 of the present invention is allowable.

Further, claims 2-7 and 25 depend directly or indirectly depend from claim 1. Claim 2 includes another limitation that the processor is used to process to process software code of the

software object. Claim 2 contains the novel limitations of claim 1 in addition to the processor processing the software code, and therefore, at least for the reasons cited herein, claim 2 is allowable. Claim 3, adds the limitation of assigning a security level to a portion of a memory to the limitations of claim 1, which as described above, are elements that are allowable for at least the reasons cited above. Claims 4 calls for adding the limitations relating to performing a virtual address memory access using a secondary table and at least one virtual memory table, to the limitations of claim 1, which as described above, are elements that are allowable for at least the reasons cited above. Claim 5 adds the limitation of adding the limitations relating to dividing a physical memory and determining at least one segment to omit from the secondary table, to the limitations of claim 1, which as described above, are elements that are allowable for at least the reasons cited above. Claim 6 adds the limitation of verifying a match between an execution security level to a security level associated with a memory segment, to the limitations of claim 1, which as described above, are elements that are allowable for at least the reasons cited above. Claim 7 adds the limitation of defining a current security level based upon a segment being executed, to the limitations of claim 1, which as described above, are elements that are allowable for at least the reasons cited above. Claim 25 adds the limitation of executing a function of said object based upon said virtual address based memory access, to the limitations of claim 1, which as described above, are elements that are allowable for at least the reasons cited above.

Similarly, claim 8 calls for a method that provides for the memory access using a virtual address, wherein the access includes utilizing a secondary table, as well as at least one virtual memory table. As described above, ***McKee*** simply does not disclose a secondary table and the Examiner's usage of the TLB is erroneous since data is written into the TLB; wherein the memory access called for by claim 8 of present invention uses information in the secondary table

as well as a virtual memory table. Therefore, ***McKee*** does not teach, disclose, or suggest all of the elements of claim 8 of the present invention. Further, claims 9-11, which depend from claim 8, respectively add limitations that include a processor being used to process software code, assigning a security level relating to a portion of a memory and defining a current security level based upon determining a segment being executed. Since claims 9-11 contain subject matter from claim 8, which is allowable, claims 9-11 are also allowable for at least the reasons cited above.

Claim 12 calls for an apparatus that includes means for performing a virtual address based memory access using said security level using a secondary table and at least one virtual memory table. As described above, ***McKee*** does not disclose a secondary table, nor does it disclose a virtual address memory access using a secondary table and a virtual table access. Therefore, ***McKee*** does not teach, disclose, or suggest all of the elements of claim 12 of the present invention.

Claim 13 calls for an apparatus that includes a memory access interface that is adapted to provide a process a virtual address based access of a portion of a memory unit using a secondary table and at least one virtual memory table to perform the virtual address based access. As described above, ***McKee*** does not disclose a virtual address based memory access using said security level using a secondary table and at least one virtual memory table. Therefore, ***McKee*** does not teach, disclose, or suggest all of the elements of claim 13 of the present invention. Further, claims 14-16, which depend from claim 13, respectively add limitations that include microprocessor, a virtual memory table, and the memory unit being comprises of a magnetic tape memory, a flash memory, a random access memory, and/or a memory residing on a

semiconductor chip. Since claims 14-16 contain subject matter from claim 13, which is allowable, claims 14-16 are also allowable for at least the reasons cited above.

Claim 17 calls for a computer readable program storage device encoded with instructions that, when executed by a computer, performs a method that includes performing a virtual address based access using a secondary table and at least one virtual memory table. As described above, ***McKee*** does not disclose a virtual address based memory access using said security level using a secondary table and at least one virtual memory table. Therefore, ***McKee*** does not teach, disclose, or suggest all of the elements of claim 17 of the present invention. Further, since claims 18-23 and 26 directly or indirectly contain subject matter from claim 17, which is allowable, claims 18-23 and 26 are also allowable for at least the reasons cited above.

Independent claims 1, 8, 12, 13, and 17are allowable for at least the reasons cited above. Additionally, dependent claims 2-7 & 25; 9-11; 14-16 & 26; and 18-23, which respectively depend from claims 1, 8, 12, 13, and 17 are also allowable for at least the reasons cited above.

Appellants acknowledge and appreciate that the Examiner asserted that claims 5 and 21 contained allowable subject matter.

VIII. CLAIMS APPENDIX

The claims currently under consideration, *i.e.,* claims 1-23 and 25-26, are listed in the Claims Appendix attached hereto.

IX. EVIDENCE APPENDIX

There is no evidence relied upon in this Appeal with respect to this section.

X. RELATED PROCEEDINGS APPENDIX

There are no related appeals and/or interferences that might affect the outcome of this proceeding.

In view of the foregoing, it is respectfully submitted that the Examiner erred in not allowing all claims (claims 1-23 and 25-26) pending in the present application over the prior art of record.

If for any reason the Examiner finds the application other than in condition for allowance, **the Examiner is requested to call the undersigned attorney at the Houston, Texas telephone number (713) 934-4069** to discuss the steps necessary for placing the application in condition for allowance.

Respectfully submitted,

WILLIAMS, MORGAN & AMERSON, P.C.
CUSTOMER NO. 23720

Date: November 2, 2006

By: /Jaison C. John/
Jaison C. John, Reg. No. 50,737
10333 Richmond, Suite 1100
Houston, Texas 77042
(713) 934-4069
(713) 934-7011 (facsimile)
ATTORNEY FOR APPELLANTS

CLAIMS APPENDIX

1. (Previously Amended) A method, comprising:

executing a software object;

establishing a security level for said software object; and

performing a virtual address based memory access using said security level, performing said virtual address based memory access comprising using a secondary table and at least one virtual memory table.

2. (Original) The method described in claim 1, wherein executing a software object further comprises using a processor to process software code of said software object.

3. (Original) The method described in claim 1, wherein establishing a security level for said software object further comprises assigning a security level relating to a memory access of at least a portion of a memory.

4. (Previously Amended) The method described in claim 1, wherein performing said virtual address based memory access using at least one of said security level further comprises:

establishing said secondary table;

receiving a memory access request based upon executing of said software object;

performing said virtual address memory access based upon said memory access request using said secondary table and said at least one virtual memory table; and

accessing a portion of a memory based upon said virtual address memory access.

5. (Original) The method described in claim 4, wherein establishing a secondary table further comprises:

dividing a physical memory into a plurality of segments;

determining at least one of said segment to omit from said secondary table and at least one un-omitted segment;

assigning a default security level to said omitted segment;

assigning a security level to said un-omitted segment; and

correlate at least one assigned segment with a virtual memory location.

6. (Previously Amended) The method described in claim 4, wherein performing said virtual address memory access based upon said memory access request further comprises:

determining at least one security level that corresponds to a segment in said secondary table;

verifying a match between an execution security level to a security level associated with a memory_segment being accessed in response to an execution of said object;

determining a virtual memory address based upon said secondary table in response to a match between said execution security level and said security level associated with said segment being accessed; and

locating a physical memory location corresponding to a virtual memory address.

7. (Original) The method described in claim 6, wherein determining at least one security level that corresponds to said segment in said secondary table further comprises:

determining a physical address from said virtual memory table;

determining a segment being executed based upon said physical address; and

defining a current security level based upon said determining of said segment being executed.

8. (Previously Amended) A method, comprising:

executing a software object;

establishing a security level for said software object;

establishing a secondary table;

receiving a memory access request based upon said executing of said software object;

determining at least one security level that corresponds to a segment in said secondary table based upon a virtual address; and

accessing a portion of a memory based upon said security level and said virtual address, accessing said portion of said memory comprising using said secondary table and at least one virtual memory table.

9. (Original) The method described in claim 8, wherein executing a software object further comprises using a processor to process software code of said software object.

10. (Original) The method described in claim 8, wherein establishing a security level for said software object further comprises assigning a security level relating to a memory access of at least a portion of a memory.

11. (Original) The method described in claim 8, wherein determining at least one security level that corresponds to a segment in said secondary table comprises:

determining a physical address from said virtual memory table;

determining a segment being executed based upon said physical address; and

defining a current security level based upon said determining of said segment being executed.

12. (Previously Amended) An apparatus, comprising:

means for executing a software object;

means for establishing a security level for said software object; and;

means for performing a virtual address based memory access using said security level, means for performing said virtual address based memory access includes, means for using a secondary table and at least one virtual memory table.

13. (Previously Amended) An apparatus, comprising:

a processor coupled to a bus;

means for coupling at least one software object to said processor;

a memory unit; and

a memory access interface coupled to said bus and said memory unit, said memory access interface to provide said processor a virtual address based access of at least a portion of said memory unit based upon at least one security level, in response to said processor executing said software object, said processor to use a secondary table and at least one virtual memory table to perform said virtual address based access.

14. (Original) The apparatus of claim 13, wherein said processor comprises at least one microprocessor.

15. (Previously Amended) The apparatus of claim 13, wherein said memory access interface comprises said virtual memory table coupled with said secondary table, said memory access interface to provide a virtual memory addressing scheme to access at least one portion of said memory unit based upon a security level.

16. (Original) The apparatus of claim 13, wherein said memory unit comprises at least one of a magnetic tape memory, a flash memory, a random access memory, and a memory residing on a semiconductor chip.

17. (Previously Amended) A computer readable program storage device encoded with instructions that, when executed by a computer, performs a method, comprising:

executing a software object;

establishing a security level for said software object; and;

performing a virtual address based memory access using said security level, performing said virtual address based memory access comprising using a secondary table and at least one virtual memory table.

18. (Original) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 17, wherein executing a software object further comprises using a processor to process software code of said software object.

19. (Original) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 17, wherein establishing a security level for said software object further comprises assigning a security level relating to a memory access of at least a portion of a memory.

20. (Previously Amended) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 17, wherein performing a virtual address based memory access using at least one of said security level further comprises:

establishing said secondary table;

receiving a memory access request based upon executing of said software object;

performing a virtual address memory access based upon said memory access request using said secondary table and said at least one virtual memory table; and

accessing a portion of a memory based upon said virtual address memory access.

21. (Original) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 20, wherein establishing a secondary table further comprises:

dividing a physical memory into a plurality of segments;

determining at least one of said segment to omit from said secondary table and at least one un-omitted segment;

assigning a default security level to said omitted segment;

assigning a security level to said un-omitted segment; and

correlate at least one assigned segment with a virtual memory location.

22. (Previously Amended) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 20, wherein performing a virtual address memory access based upon said memory access request further comprises:

determining at least one security level that corresponds to a segment in said secondary table;

verifying a match between an execution security level to a security level associated with a memory segment being accessed in response to an execution of said object;

determining a virtual memory address based upon said secondary table in response to a match between said execution security level and said security level associated with said segment being accessed; and

locating a physical memory location corresponding to a virtual memory address.

23. (Original) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 22, wherein determining at least one security level that corresponds to a segment in said secondary table comprises:

determining a physical address from said virtual memory table;

determining a segment being executed based upon said physical address; and

defining a current security level based upon said determining of said segment being executed.

24. (Canceled)

25. (Previously Presented) The method described in claim 1, further comprising executing a function of said object based upon said virtual address based memory access.

26. (Previously Presented) The computer readable program storage device encoded with instructions that, when executed by a computer, performs the method described in claim 17, the method further comprising executing a function of said object based upon said virtual address based memory access.